Installing python

Windows

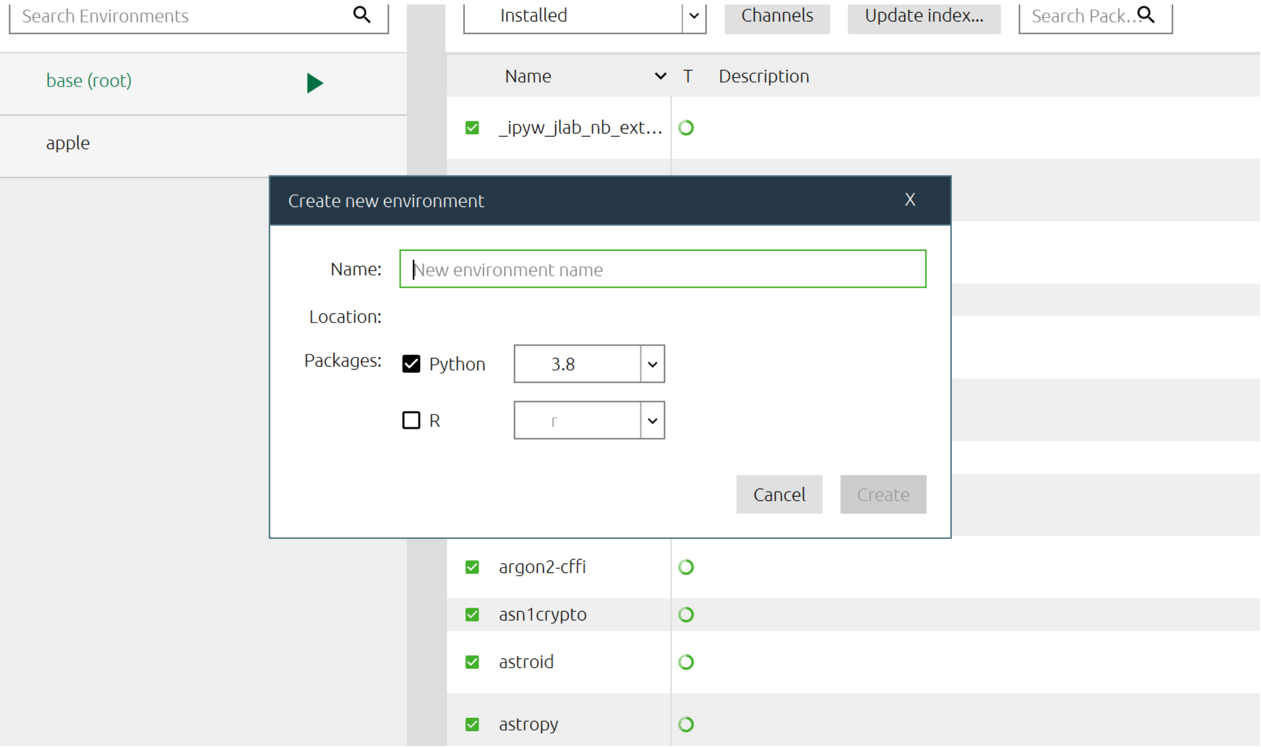
1. ***Install Anaconda***

> Anaconda includes both Python and conda, and additionally bundles a suite of other pre-installed packages geared toward scientific computing. Because of the size of this bundle, expect the installation to consume several gigabytes of disk space.

> It includes some data science relative language like python and R.

> It's convenient to create a vitual env and install the package.

- **Download**: click Anaconda Installation and install the Anaconda.
- **Create new environment**: Find the "*Environments*" on the left and click the "*Create*". Pick a name you like than tick the **python 3.8** in the "*package*" option.



Create new environment

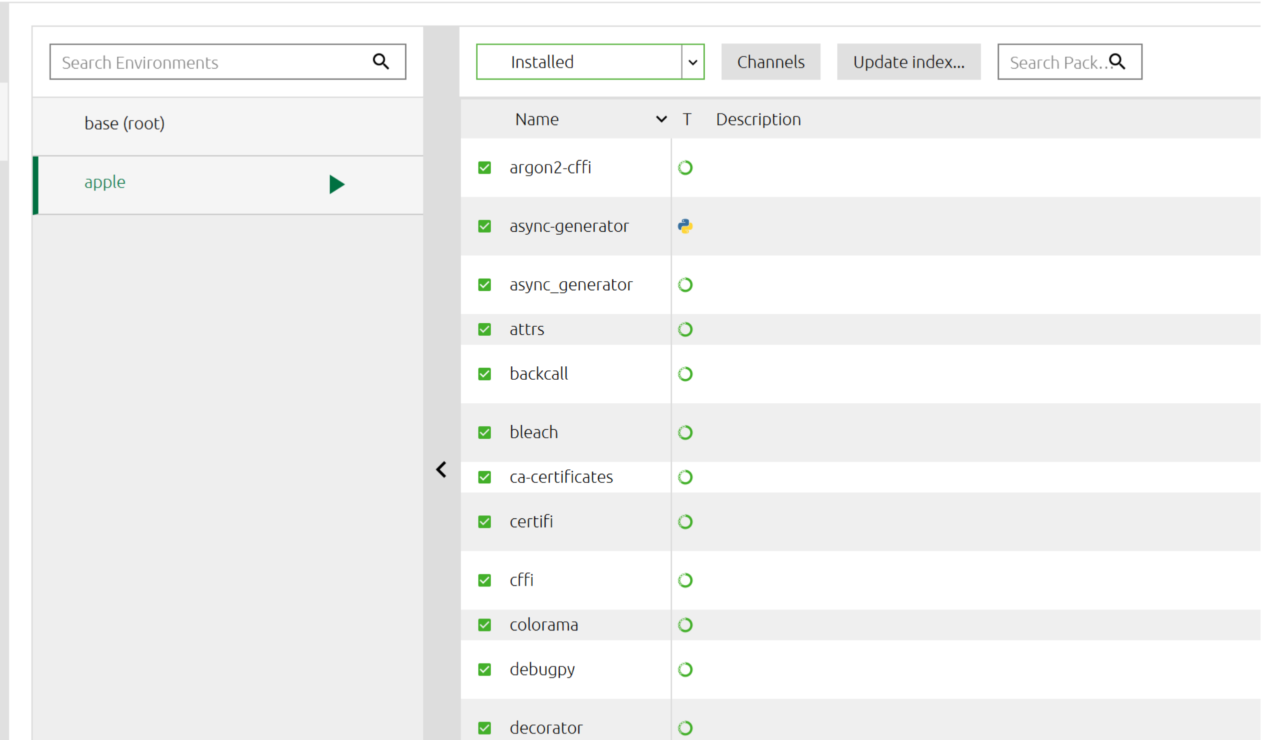
Name: New environment name

Location:

Packages: ☑ Python 3.8

□ R r

- **Congratulations!** Now you have your own visual environment.



	Name	T	Description
☑	argon2-cffi	○	
☑	async-generator	🐍	
☑	async_generator	○	
☑	attrs	○	
☑	backcall	○	
☑	bleach	○	
☑	ca-certificates	○	
☑	certifi	○	
☑	cffi	○	
☑	colorama	○	
☑	debugpy	○	
☑	decorator	○	

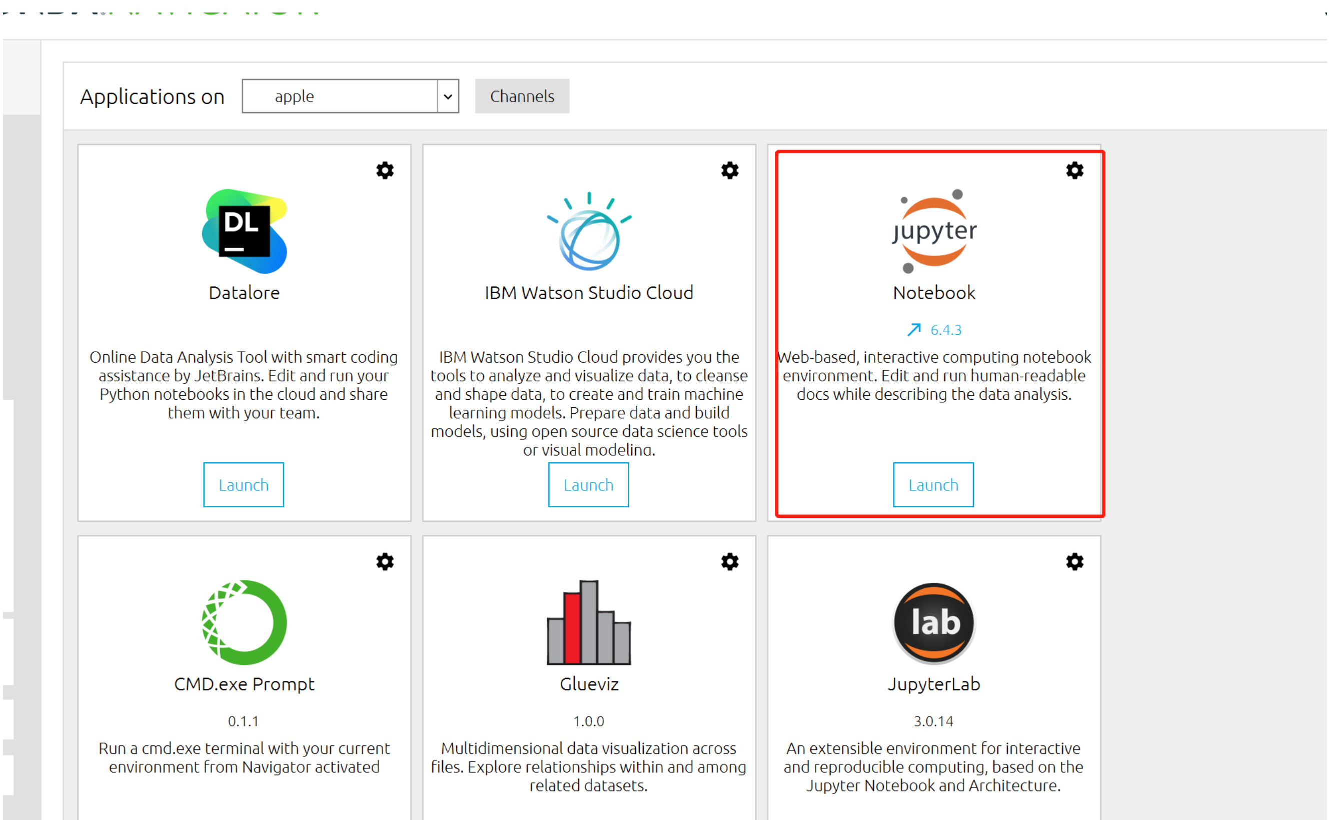
2. ***Install Jupyter Notebook***

> The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

> Very friendly to beginners.

> Simple interface and operation and debugging are very simple.

- **Install/Launch**: back to "*Home*". Then choose "*Applications on*" "*your_env_name*". Now you can find the **Jupyter Notebook** and install/launch it.



Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Run a cmd.exe terminal with your current environment from Navigator activated

Multidimensional data visualization across files. Explore relationships within and among related datasets.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

- **Coding**: Create a new python script and enjoy it!!!

3. ***Modules Installation***

> A **module** allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

- Method 1: Install module by using **pip install** in your cmd.

```
# Example
pip install numpy
pip install pandas
# or
pip3 install numpy
pip3 install pandas
```

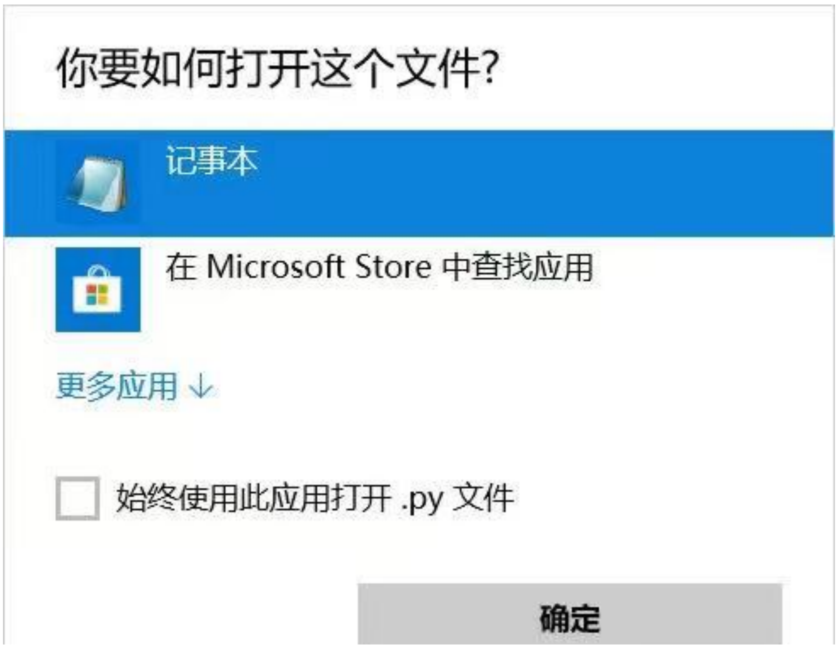
- Method 2: Find the modules in your **Anaconda Environment**.

Note: In Windows, the default path in Jupiter notebook is C:/. If you need to change it, use these command:

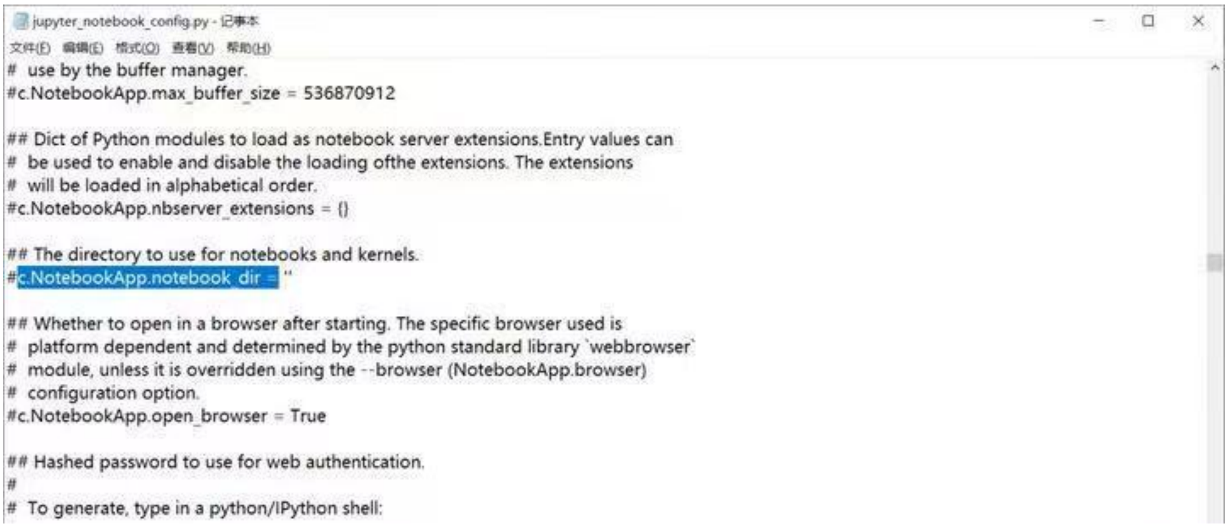
1.

```
jupyter notebook --generate-config
```

2. Find the doc we just create "Jupiter_notebook_config.py" and open it with note.

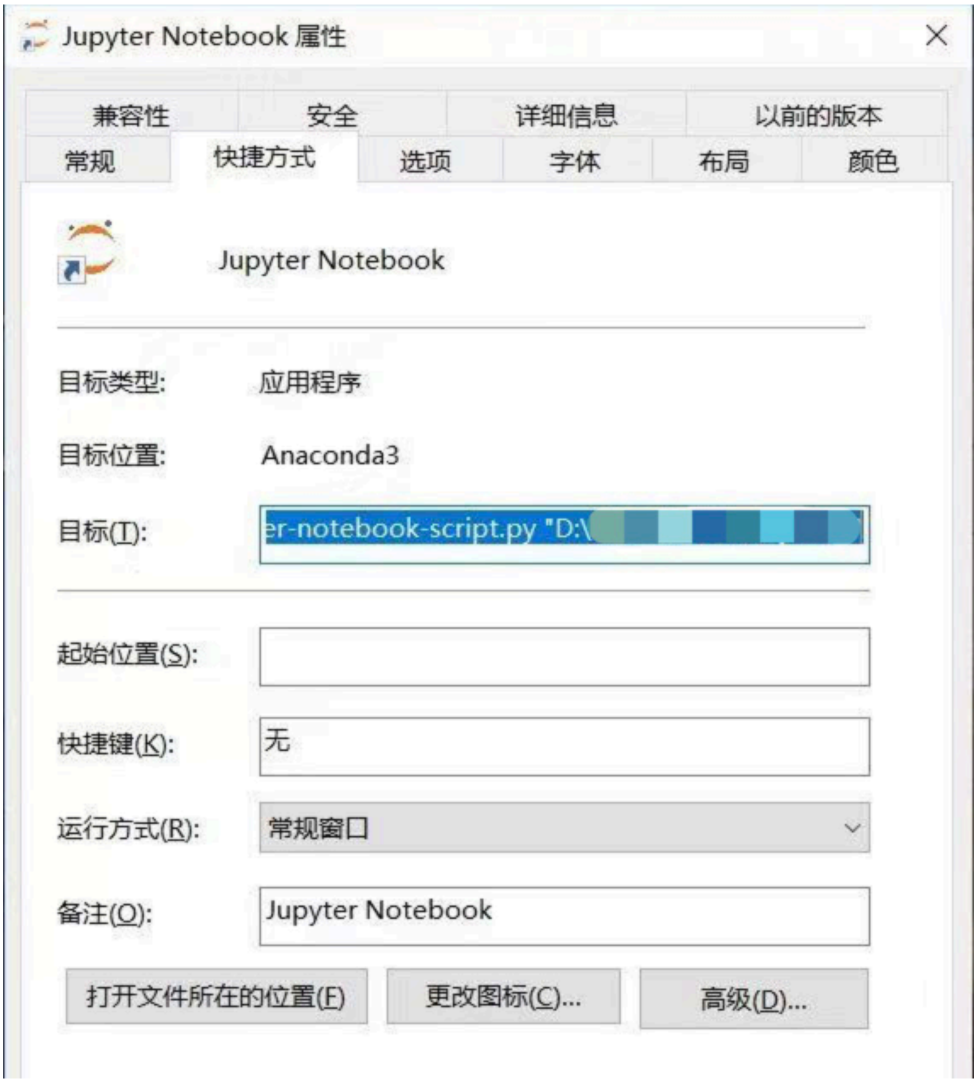

你要如何打开这个文件?

3. "Ctrl + F" to find following string, change the '' with your path and delete the #.


jupyter_notebook_config.py - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
#  use by the buffer manager.
#c.NotebookApp.max_buffer_size = 536870912

## Dict of Python modules to load as notebook server extensions.Entry values can
#  be used to enable and disable the loading ofthe extensions. The extensions
#  will be loaded in alphabetical order.
#c.NotebookApp.nbserver_extensions = {}

## The directory to use for notebooks and kernels.
#c.NotebookApp.notebook_dir = ''

## Whether to open in a browser after starting. The specific browser used is
#  platform dependent and determined by the python standard library `webbrowser`
#  module, unless it is overridden using the --browser (NotebookApp.browser)
#  configuration option.
#c.NotebookApp.open_browser = True

## Hashed password to use for web authentication.
#
#  To generate, type in a python/IPython shell:
```

4. Lastly, modify the path of Jupyter notebook shortcut. Replace the %USERPROFILE% with your path.



Jupyter Notebook

目标类型: 应用程序

目标位置: Anaconda3

目标(T): er-notebook-script.py "D:\[redacted]

起始位置(S):

快捷键(K): 无

运行方式(R): 常规窗口

备注(O): Jupyter Notebook

MacOS

1. ***Install Homebrew***

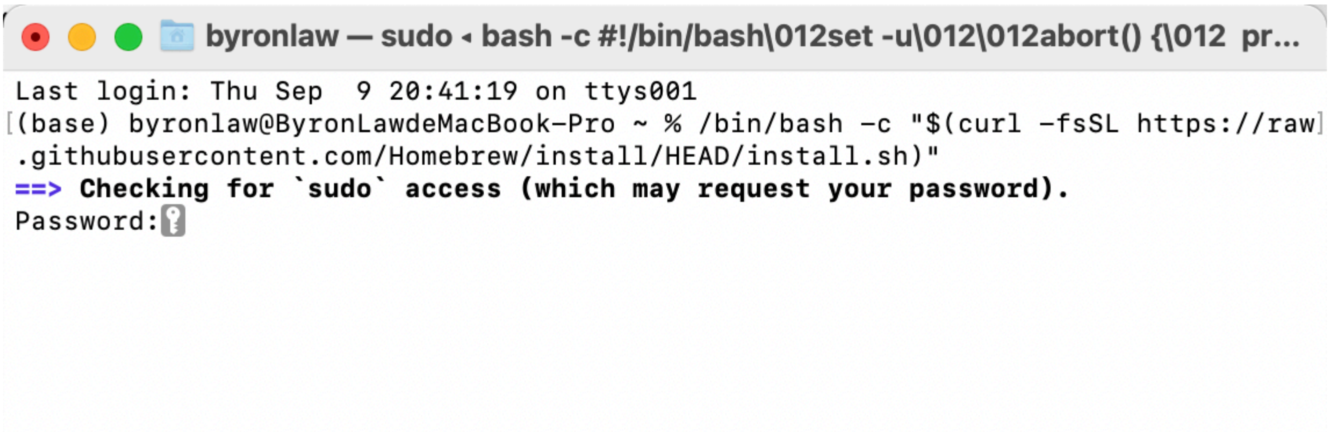
> Homebrew installs the stuff you need that Apple (or your Linux system) didn’t.

> Homebrew installs packages to their own directory and then symlinks their files into `/usr/local`.

> Homebrew won’t install files outside its prefix and you can place a Homebrew installation wherever you like.

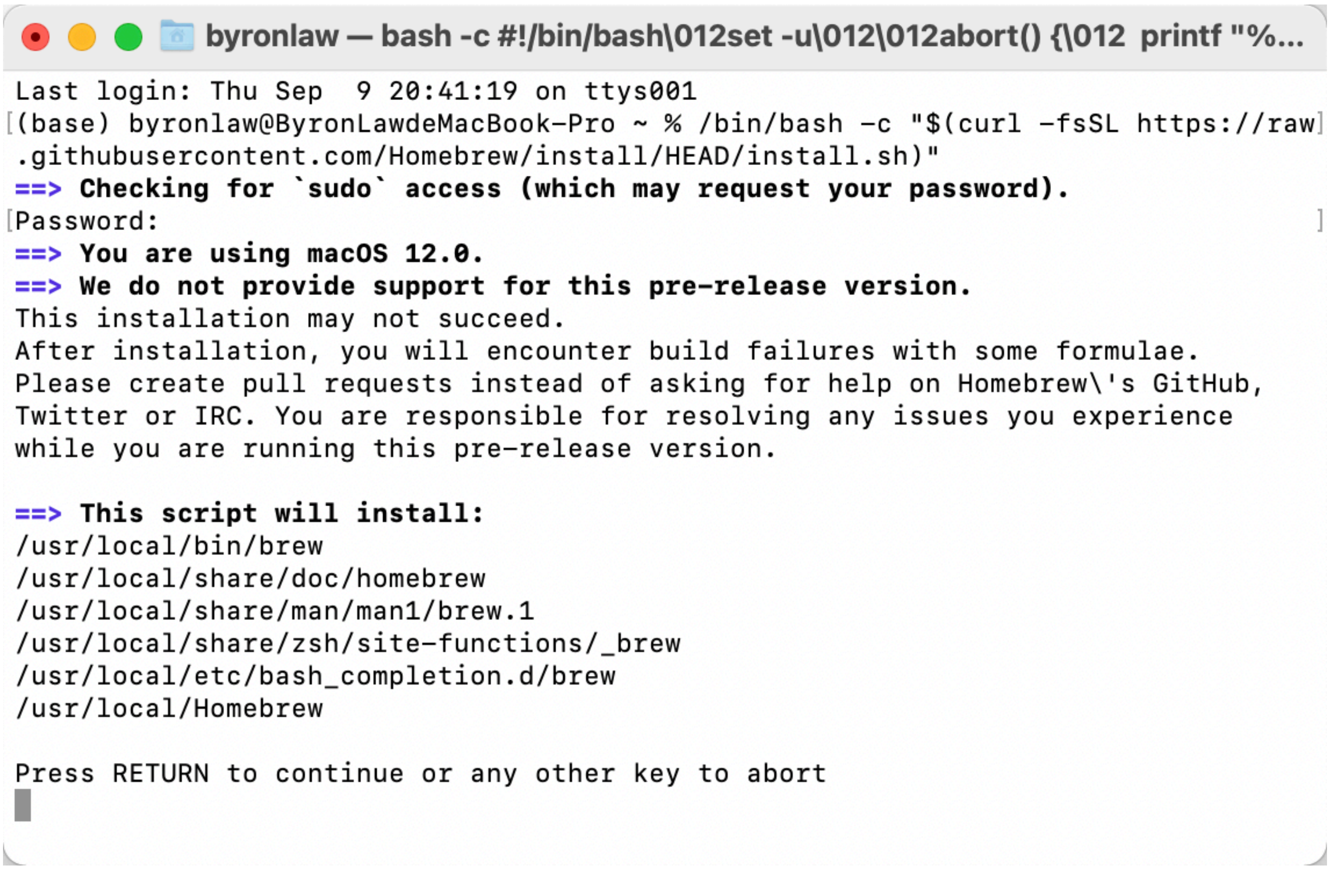
```
# Install Homebrew
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Your terminal will ask for Super User-level access. You will need to type your password to run this command. This is the same password you type when you log into your Mac. Type it and hit enter.


byronlaw — sudo ◂ bash -c #!/bin/bash\012set -u\012\012abort() {\012 pr...

```
Last login: Thu Sep  9 20:41:19 on ttys001
(base) byronlaw@ByronLawdeMacBook-Pro ~ % /bin/bash -c "$(curl -fsSL https://raw
.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password).
Password:
```

- Homebrew will ask you to confirm you want to install the following. You have to press enter to continue. (Or press any other key if you get cold feet.)


byronlaw — bash -c #!/bin/bash\012set -u\012\012abort() {\012 printf "%...

```
Last login: Thu Sep  9 20:41:19 on ttys001
(base) byronlaw@ByronLawdeMacBook-Pro ~ % /bin/bash -c "$(curl -fsSL https://raw
.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password).
Password:
==> You are using macOS 12.0.
==> We do not provide support for this pre-release version.
This installation may not succeed.
After installation, you will encounter build failures with some formulae.
Please create pull requests instead of asking for help on Homebrew\'s GitHub,
Twitter or IRC. You are responsible for resolving any issues you experience
while you are running this pre-release version.

==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
```

2. ***Install pyenv to Manage Your Python Versions***

> Now let's take a moment to install PyEnv. This library will help you switch between different versions of Python (in case you need to run Python 2.x for some reason, and in anticipation of Python 4.0 coming).

```
brew install pyenv
```

3. ***Install python or update your python version***

> Note that you can substitute 3.9.2 for whatever the latest version of Python is.

```
# Install python
pyenv install 3.9.2
```

Now you got the python and welcome to the python world!

```
# Check the python version, pip version and pip list
python --version # python3 --version
pip --version # pip3 --version
pip list # pip3 list
```

4. ***Install Jupyter Notebook***

   > Use **pip** to install the jupyter notebook

   ```
   pip install notebook
   ```

-About Modules-

1. ***Numpy***

   > **NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

2. ***Pandas***

   > **pandas** is a software librarywritten for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

3. ***Matplotlib***

   > **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

4. ***Scikit-learn***

   > **Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a free software machine learning for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5. ***Seaborn***

> **Seaborn** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

6. ***Beautifulsoup***

> **Beautiful Soup** is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.